

(51) International Patent Classification ⁵ : G06F 15/18	A1	(11) International Publication Number: WO 93/21586 (43) International Publication Date: 28 October 1993 (28.10.93)
---	-----------	--

Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

BNSDOCID: <WO_____9321586A1 | >

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	FR	France	MR	Mauritania
AU	Australia	GA	Gabon	MW	Malawi
BB	Barbados	GB	United Kingdom	NL	Netherlands
BE	Belgium	GN	Guinea	NO	Norway
BF	Burkina Faso	GR	Greece	NZ	New Zealand
BG	Bulgaria	HU	Hungary	PL	Poland
BJ	Benin	IE	Ireland	PT	Portugal
BR	Brazil	IT	Italy	RO	Romania
CA	Canada	JP	Japan	RU	Russian Federation
CF	Central African Republic	KP	Democratic People's Republic of Korea	SD	Sudan
CG	Congo	KR	Republic of Korea	SE	Sweden
CH	Switzerland	KZ	Kazakhstan	SK	Slovak Republic
CI	Côte d'Ivoire	LI	Liechtenstein	SN	Senegal
CM	Cameroon	LK	Sri Lanka	SU	Soviet Union
CS	Czechoslovakia	LU	Luxembourg	TD	Chad
CZ	Czech Republic	MC	Monaco	TG	Togo
DE	Germany	MG	Madagascar	UA	Ukraine
DK	Denmark	ML	Mali	US	United States of America
ES	Spain	MN	Mongolia	VN	Viet Nam
FI	Finland				

DESCRIPTIONAutonomous Learning And Reasoning Agent
Cross-Reference To Related ApplicationTitle of the Invention

This application is a continuation-in-part of copending application Serial No. 07/ 664,561, filed March 4, 1991 in the name of inventors Bradley P. Allen and S. Daniel Lee and titled "CASE-BASED REASONING SYSTEM", hereby incorporated by reference as if fully set forth herein.

Background of the Invention1. Field of the Invention

10 This invention relates to case-based reasoning and to a case-based reasoning system which performs autonomous learning in a real-world environment.

2. Description of Related Art

15 While computers are capable of tremendous processing power, their ability to use that processing power for reasoning about complex problems has so far been limited. Generally, before a computer can be used to address a complex problem, such as one which requires the attention of a human expert, it has been necessary to distill the knowledge of that expert into a set of inferential rules (a "rule base") which allow an automated processor to reason in a limited field of application. While this method has been effective in some cases, it has the natural drawback that it often requires a substantial amount of time and effort, by both computer software engineers and experts in the particular field of application, to produce a useful product.

Moreover, rule-based systems of this type present a difficult programming task. Unlike more prosaic programming tasks, constructing a rule base is sometimes

counterintuitive, and may be beyond the ability of many application programmers. And once a rule-based system has been constructed based on the knowledge of a human expert, it may be difficult to accommodate changes in the field of operation in which the processor must operate. Such changes might comprise advances in knowledge about the application field, additional tasks which are intended for the processor, or changes in or discoveries about the scope of the application field.

One proposed method of the prior art is to build automated reasoning systems which operate by reference to a set of exemplar cases (a "case base"), to which the facts of a particular situation (the "problem") may be matched. The processor may then perform the same action for the problem as in the exemplar case. While this proposal has been well-received, case-based systems of this type may still require a substantial amount of human effort to identify exemplar cases and present a processor with sufficient information that cases may be matched and acted upon. For example, it may be necessary to deduce or supply extensive information about a complex environment so as to determine a preferred set of exemplar cases.

A parent copending application, Serial No. 07/664,561, filed March 4, 1991, discloses inventions in which a case-based reasoning system is smoothly integrated into a rule-based reasoning system, and in which an automated reasoning system may dynamically adapt a case base to problems which it encounters. An aspect of the inventions disclosed in that application also includes a technique in which a system may be set to work with a limited case base, and may solicit human advice for treatment of new problems which are not already well-treated by the case base, thus learning how to do its job on a dynamic basis.

Another copending application, Serial No. _____, Lyon & Lyon Docket No. 195/018, filed the same day as this application, discloses inventions in

which a machine learning system may operate in conjunction with a relational database system, and particularly in which a machine learning system may operate in conjunction with a relational database system with an SQL interface.

5 This allows the machine learning system to use high-speed searching power of computer systems which have been designed for use with relational database systems with an SQL interface, and allows the machine learning system to be smoothly integrated into computer systems which have
10 relational databases, even if those databases were not designed to work with learning or reasoning systems of any kind.

It would be advantageous if an automated reasoning system could dynamically create its own case base in
15 response to problems which it encounters, thus learning how to do its job on a dynamic basis and without substantial human intervention, or at least with only occasional human intervention. Limited intervention would allow an automated reasoning system to examine a larger
20 set of cases and to determine a preferred set of exemplar cases without an external agent, such as a human operator, having to deduce or supply extensive information about a complex environment.

It would also be advantageous if an automated
25 reasoning system could operate autonomously in a complex environment, possibly with external intervention such as positive or negative reinforcing stimuli. External stimuli might be in response to a result of the system's attempts to manipulate its environment, or might be
30 provided by an external agent, such as a human operator. Accordingly, it is an object of the invention to provide an automated reasoning system which does not require intervention for every case.

Summary of the Invention

35 The invention provides a software agent which performs autonomous learning in a real-world environment.

The autonomous agent may learn by reinforcement (including positive and negative, and delayed and sporadic, reinforcement), in addition to learning by example and learning by being told what to do. In a preferred embodiment, the autonomous agent may be implemented in a case-based reasoning system, which may be coupled to a sensor for gathering information from, and to an effector for manipulating, its environment (which may comprise a software environment, a physical environment, or some combination thereof). In addition to gathering a case base of experience in its environment, the autonomous agent may tune that case base in response to an evaluation of how well it is operating in that environment. The evaluation may be its own, or may be in response to a stimulus such as a reward or punishment. In addition to taking action on a problem based on its case base, the autonomous agent may take action to gather information so as to determine which cases are most appropriate to that problem.

In a preferred embodiment, the autonomous agent may comprise a memory of cases, the contents of that memory being determined by a genetic technique for producing, evaluating and selecting cases. New cases may be produced by inspection of scenarios from the environment, by mutation of old cases, or by combining or selecting features from old cases; thus the memory may comprise cases the autonomous agent has never encountered in the environment. The stored cases may be evaluated in response to a history of previous matches and in response to an external stimulus, and evaluations (such as measures of accuracy and utility) may be associated with stored cases. The contents of the memory may be limited to a set of cases which provides a preferred model of the environment, such as those cases which have the better evaluations.

In a preferred embodiment, the autonomous agent may comprise a selection technique based on multiple factors,

such as match quality, case accuracy, or case utility. The selection technique may also induce experimentation by the autonomous agent, such as by employing a random or pseudorandom effect in selecting cases. The selection
5 technique may also distinguish between those actions which solve problems and those actions which gather further information so as to better solve problems. Multiple autonomous agents may form a collective entity and may cooperate to select an action to be performed by that
10 collective entity.

Brief Description of the Drawings

Figure 1 shows a block diagram of an autonomous agent embedded in a complex environment.

Figure 2 shows a block diagram of a behavior module
15 of an autonomous agent.

Figure 3 shows a data flow diagram of a genetic technique for producing, evaluating and selecting cases.

Figure 4 shows a data flow diagram of a technique for selecting cases in a memory.

Figure 5 shows a block diagram of an intelligent
20 office equipment device including an autonomous agent.

Figure 6 shows a block diagram of a customer service system with a help-desk system including an autonomous agent.

Figure 7 shows a block diagram of a knowledge
25 discovery system including an autonomous agent.

Appendix A shows an example software environment and autonomous agent for distinguishing between classes of irises.

30 Description of the Preferred Embodiment

An embodiment of this invention may be used together with inventions which are disclosed in a copending application titled "MACHINE LEARNING WITH A RELATIONAL DATABASE", application Serial No. _____, Lyon
35 & Lyon Docket No. 195/018, filed the same day in the name

of the same inventor, hereby incorporated by reference as if fully set forth herein.

Figure 1 shows a block diagram of an autonomous agent embedded in a complex environment.

5 A software agent 101 may be embedded in an environment 102 so that the agent 101 may receive a stimulus 103 from the environment 102 by receiving a stimulus message 104, and may perform an action 105 which affects the environment 102 by sending an action message
10 106. In a preferred embodiment, the stimulus message 104 and the action message 106 may each comprise a manipulable software object 107 and may transmit that software object 107 from source to destination. Software objects 107 may
15 objects 107 as is well known in the art. Object-oriented systems are more fully described in "Object-Oriented Design With Applications" by Grady Brooch, published by Benjamin/Cummings Publishing, Redwood City, California (1991), hereby incorporated by reference as if fully set
20 forth herein.

In a preferred embodiment, the environment 102 may comprise either a software environment or a physical environment, or some combination thereof. For example, the environment 102 may comprise a physical room; the
25 agent 101 may receive a LIGHTS stimulus message 104 telling if a set of lights in the room is off or on; and the agent 101 may send a LIGHTS-ON action message 106 to turn on the lights. Alternatively, the environment 102 may comprise a graphic database; the agent 101 may receive
30 a PICTURE stimulus message 104 telling about a picture in the database; and the agent 101 may send an ADD-PROPERTY action message 106 to add a property value (e.g., "author = Vermeer") to the picture.

A set of receptors 109 may receive the stimulus
35 message 104 and generate a features message 110. In a preferred embodiment, the receptors 109 may map a set of aspects or features 111 of the environment 102 (especially

those relating to external stimuli) into a set of feature objects 112, which may comprise software objects 107 which describe the aspects or features 111 of the environment 102. For example, the environment 102 may comprise an oil refinery and the feature objects 112 may comprise temperature and pressure values at various positions in the refinery.

The set of receptors 109 may also generate a reward message 113, which may comprise a scalar-valued reinforcement 114 which measures what success the agent 101 is having with the environment 102. In a preferred embodiment, the value of the reinforcement 114 may be determined by the receptors 109, or may be externally supplied, such as from a human operator 115. For example, the environment 102 may comprise a loan portfolio database and the reinforcement 114 may measure loan performance, or the environment 102 may comprise a music audio database and the reinforcement 114 may be entered by the human operator 115 after listening to selections made by the agent 101.

A set of proprioceptors 116 may generate a motives message 117, which may comprise software objects 107 which describe goals or needs of the agent 101. In a preferred embodiment, the proprioceptors 116 may map aspects or features of the environment 102 (especially those relating to internal states of the agent 101 itself) into those goals or needs. The goals or needs of the agent 101 itself are thus a kind of feature 111 which the agent 101 may consider, and are similar to the features 111 which are reported by the receptors 109. For example, the environment 102 may comprise a piece of office equipment such as a photocopier and the motives message 117 may comprise internally generated features 111 such as the need for new ink or paper. The proprioceptors 116 may be altered by a designer so as to change the essential goals of the agent 101.

A behavior module 118 may receive the motives message 117, the features message 110 and the reward message 113, and may generate a queries message 119, which may comprise software objects 107 which describe a request 120 by the agent 101 for further information, and a commands message 121, which may comprise software objects 107 which describe a command 122 by the agent 101 to affect the environment 102. In a preferred embodiment, the behavior module 118 may comprise one like that disclosed with figure 2.

A set of effectors 123 may receive the queries message 119 and the commands message 121 and generate the action message 106. In a preferred embodiment, the action message 106 may be coupled to the environment 102 (e.g., a physical device or another software element) and may cause an effect in the environment 102. For example, the environment 102 may comprise a chess program and the action message 106 may direct the chess program to make a particular move on behalf of the agent 101.

In a preferred embodiment, the software agent 101 may be implemented with an automated processor 124, which may execute a software inference engine 125 for reasoning using a set of cases 126 in a case base 127 and a set of rules 128 in a rule base 129. In a preferred embodiment, the processor 124 may comprise a system having a processor, memory comprising a stored program, memory comprising data, and input/output devices, as is well known in the art. The operation and software structures of this system are described herein in terms of their functions and at a level of detail which would be clear to those of ordinary skill in the art. It would be clear to anyone of ordinary skill in the art, after perusal of the specification, drawings and claims herein, that modification and/or programming (using known programming techniques) of a processor of known design to achieve these functions would be a straightforward task and would not require undue experimentation.

In a preferred embodiment, the processor 124 may comprise an IBM-compatible PC configured to be able to execute the MicroSoft Windows 3.0 and DOS 3.1 software, and having a hard disk drive, a mouse, and a VGA display.

5 At least a 286 processor with four megabytes of memory is preferred; a 386 processor with eight megabytes of memory is more preferred. The MicroSoft Windows 3.0 software is preferably executed in 386 enhanced mode.

Figure 2 shows a block diagram of a behavior module
10 of an autonomous agent.

A memory module 201 may receive the motives message 117, the features message 110 and the reward message 113, and may generate a set of new cases 202, a set of case update messages 203, and a set of case retrieval messages
15 204. The new cases 202 may be transmitted to a case database 205 for storage, and to a case index 206 for indexing. In a preferred embodiment, aspects of a case-based reasoning system like that disclosed in parent copending application, Serial No. 07/ 664,561, filed March
20 4, 1991, consistent with the functions disclosed herein, may be used for case indexing and matching. The case database 205 may store the actual cases 126; the case index 206 may comprise a mapping from the features 111 to the cases 126 themselves.

25 In a preferred embodiment, the case database 205 may comprise a limited number of stored cases 126, the contents of the case database 205 being determined by a genetic technique for producing, evaluating and selecting the cases 126, such as a genetic technique like that
30 disclosed with figure 3. Thus, while the case database 205 might store less than all of the cases 126 which the autonomous agent 101 has encountered, it might maintain a set of cases 126 which provides a preferred model of the environment, such as those cases 126 which allow the agent
35 101 to distinguish variant problem scenarios and to act autonomously and intelligently in those variant problem scenarios.

The case index 206 may comprise an index of the cases stored in the case database 205 (e.g., a set of case identifiers 207), organized so that cases 126 may be matched and retrieved, and may respond to the case retrieval message 204 by providing a matches message 208. The matches message 208 may comprise the case identifiers 207, and other information which a selector 209 may require.

The selector 209 may receive the matches message 208 and may also receive a cases message 210 from the case database 205, and may generate the queries message 119 and the commands message 121. In a preferred embodiment, the selector 209 may employ a technique for selecting cases like that disclosed with figure 4. The cases message 210 may comprise data from the cases 126, including evaluation data such as accuracy and utility values.

Figure 3 shows a data flow diagram of a genetic technique for producing, evaluating and selecting cases.

The case database 205 may comprise its current set of cases 126, each of which may comprise a features element 301, which may generally indicate when the case 126 may be useful, and an action element 302, which may indicate the action 105 to take and an evaluation 303 of that action 105. In a preferred embodiment, the evaluation 303 may comprise an accuracy value 304, which may indicate how "good" the case 126 generally is when used, and a utility value 305, which may indicate how "often" the case 126 generally is usable. For example, if the environment 102 is a carpenter's bench, the utility value 305 of a hammer may be high, because it is often usable, even though its accuracy value 304 indicates that, even when the hammer is usable, it is not always the best choice. Similarly, the utility value 305 of a plane may be low, because it is only used in specialized situations, even though its accuracy value 304 indicates that whenever it is usable, it is the correct tool to choose.

An evaluation module 306 may receive the reward message 113 and a history message 307 (indicating the history of matches), and may operate on the cases 126 to adjust their evaluations 303, particularly their accuracy values 304 and their utility values 305. The evaluation module 306 may respond to the reward message 113 by altering the utility values 305 of the cases 126 to "reinforce" those cases 126 which correspond to the action which resulted in the reinforcement 114. Thus, rewards are "credited" to the cases 126 which earned them. Moreover, the evaluation module 306 may also alter the utility values 305 of those cases 126 which correspond to the action just previous to the reinforced action as well. Thus, rewards are also credited to the cases 126 which "led up to" them.

In a preferred embodiment, the evaluation module 306 may alter the utility value 305 of each case 126 by adding the reinforcement 114 and the utility value 305 of the case 126 which is the "best match" for the next action. Thus:

$$\text{utility value (time } t) = \text{reinforcement} + \text{utility value (time } t+1)$$

A reproduction module 308 may operate on the cases 126 to adjust their features elements 301 or the action 105 in their action elements 302, by one or more of several techniques. For example, the reproduction module 308 may create and delete cases 126. The reproduction module 308 may create cases 126, for example by inserting new cases 126 into the case database 205 as they are encountered. One method for such instance-based learning is disclosed in parent copending application, Serial No. 07/ 664,561, filed March 4, 1991. The reproduction module 308 may delete cases 126, for example when their accuracy values 304 or their utility values 305 fall below a threshold.

The reproduction module 308 may also make new cases 126 from the old cases 126 in the case database 205. The

reproduction module 308 may mutate cases 126, for example by altering one or more features 111 in one case 126 in the case database 205. The reproduction module 308 may also cross-pollinate cases 126, for example by selecting
5 some features 111 from one old case 126 and some features 111 from another old case 126 to create one or more new cases 126.

The operation of the evaluation module 306 and the reproduction module 308 serve to continually review and
10 update the selection of cases 126 in the case database 205, so that the case database 205 is continually altered into a progressively better set of cases 126. The genetic technique is influenced by the reward message 113, which provides a form of evolutionary pressure towards selection
15 of those sets of cases 126 in the case database 205 which maximize rewards.

Figure 4 shows a data flow diagram of a technique for selecting cases in a memory.

A matching module 401 may review the cases 126 in the
20 case database 205 and may generate a match table 402. In a preferred embodiment, a technique for selecting cases 126 may employ techniques for matching features 111 such as attribute-value pairs, and for generating the match table 402, like those disclosed in parent copending
25 application, Serial No. 07/ 664,561, filed March 4, 1991, consistent with the functions disclosed herein. The match table 402 may comprise a set of cases 126 (or indices of cases 126), each having a match quality value 403, the accuracy value 304 of that case 126, and the utility value
30 305 of that case 126.

A randomization module 404 may choose one of the cases 126 based on a random or pseudorandom effect. As used herein, "random" effects include pseudorandom effects and related methods which may be used to achieve similar
35 results. In a preferred embodiment, the randomization module 404 may choose one of the cases 126 in the match table 402 with a probability of choice for each case 126

which is linearly proportional to its accuracy value 304. The cases 126 in the match table 402 are therefore chosen for matching based on their match quality values 403, but they are selected for action based on their accuracy values 304. This allows the technique for selection to encourage experimentation by the agent 101.

It would be clear to one of ordinary skill in the art, after perusal of the specification, drawings and claims herein, that the randomization module 404 may employ other and further techniques for choosing one of the cases 126 in the match table 402. For example, another measure associated with the cases 126 (such as their utility value 305), or a combination of values associated with the cases 126 might be employed in place of the accuracy value 304. Moreover, the probability of choice may be other than linearly proportional to the accuracy value 304. It would be clear to one of ordinary skill in the art that such other and further techniques would be workable, and are within the scope and spirit of the invention.

Figure 5 shows a block diagram of an intelligent office equipment device including an autonomous agent.

A device 501, such as a photocopier or printer, may be coupled to a device driver interface 502, which may be coupled to a control processor 503 including the autonomous agent 101. The device 501 may of course be one of many different devices, such as a fax machine, a modem, or a telephone.

The device driver interface 502 provides the stimulus message 104 to the control processor 503, which delivers the stimulus message 104 to the autonomous agent 101. The autonomous agent 101 generates the action message 106, which the control processor 503 delivers to the device driver interface 502.

The motives message 117 may reflect a set of goals, such as eliminating or reducing the need for user involvement in maintenance of the device 501, and to

anticipate failures of the device 501 so that preventative maintenance may be performed so as to reduce failures and to increase mean-time-between-failure. The features message 110 may comprise sensor readings from the device 501, and commands from users and dealers. The action message 106 may comprise orders to the device driver interface 502 to alter settings on the device 501, and may also comprise messages to users and dealers such as diagnostic and repair messages. The reward message 113 may comprise a negative reinforcement 114 whenever the device 501 fails.

Figure 6 shows a block diagram of a customer service system with a help-desk system including an autonomous agent.

A user 601, such as a human being, may access the system by means of a telephone connection 602. A help-desk system 603 may attempt to respond to inquiries and resolve problems, with the aid of the autonomous agent 101. The help-desk system 603 provides the stimulus message 104 to the autonomous agent 101. The autonomous agent 101 generates the action message 106 for the help-desk system 603.

The help-desk system 603, its purpose and operation, may be like that shown in parent copending application, Serial No. 07/ 664,561, filed March 4, 1991. It may receive touch-tone commands from the user 601 and may generate voice messages by means of voice response units; receiving touch-tone commands and generating voice messages are well known in the art.

The motives message 117 may reflect a set of goals, such as responding to user inquiries with appropriate information and properly resolving user problems, so as to provide an intelligent help-desk system 603 which focusses on the user problem and responds to that problem, and which adapts to user behavior. The features message 110 may comprise touch-tone commands from the user 601, data indicating voice response unit messages given to the user

601, and state information about the telephone connection 602. The action message 106 may comprise commands to the help-desk system 603 or commands to a telephone system 604 controlling the telephone connection 602 (e.g., to
5 transfer the telephone connection 602 to a human operator).

The help-desk system 603 may consider that it has resolved the user problem when it reaches the end of its response or when it has transferred the telephone
10 connection 602 to a human operator. The reward message 113 may comprise a positive reinforcement 114 when the help-desk system 603 resolves the user problem, and a negative reinforcement 114 whenever the telephone connection 602 is broken by the user 601 prior to that.

15 Figure 7 shows a block diagram of a knowledge discovery system including an autonomous agent.

A knowledge database 701 may be coupled to a database interface 702, which may be coupled to a knowledge processor 703 including the autonomous agent 101. The
20 knowledge database 701 may of course be one of many different types of databases, such as an object-oriented database or a relational database.

The database interface 702 provides the stimulus message 104 to the knowledge processor 703, which delivers
25 the stimulus message 104 to the autonomous agent 101. The autonomous agent 101 generates the action message 106, which the knowledge processor 703 delivers to the database interface 702.

The motives message 117 may reflect a set of goals,
30 such as discovering useful knowledge or patterns in the data in the knowledge database 701, so as to allow the autonomous agent 101 to generalize from data and to capture useful statements about regularities in the knowledge database 701. The features message 110 may
35 comprise statements about data in the knowledge database 701, such as schema definitions or database tables. The action message 106 may comprise orders to the database

interface 702 to alter or interrogate the knowledge database 701, such as database queries and transactions. The reward message 113 may comprise a positive reinforcement 114 whenever the autonomous agent 101
5 discovers "useful" information, for some predetermined definition of "useful" (e.g., an approval message from a human operator).

Appendix A (pages ____-____) shows an example software environment and autonomous agent for
10 distinguishing between classes of irises. The example software environment comprises a "trainer" for the autonomous agent 101 which determines when the autonomous agent 101 has made a correct determination and provides an appropriate reward message 113. Some exemplary data
15 statements are also included.

Alternative Embodiments

While preferred embodiments are disclosed herein, many variations are possible which remain within the concept and scope of the invention, and these variations
20 would become clear to one of ordinary skill in the art after perusal of the specification, drawings and claims herein.

Claims

1. An autonomous learning and reasoning software agent.
2. A software agent as in claim 1, comprising means
5 for responding to reinforcement from an environment.
3. A software agent as in claim 1, comprising a case base of experience in an environment.
4. A software agent as in claim 1, comprising a case base of experience in an environment which has been
10 altered in response to an evaluation of said experience.
5. A software agent as in claim 4, wherein said evaluation may comprise a self-generated evaluation or a reinforcement from said environment.
6. An autonomous software agent, comprising
15 a sensor for gathering information from an environment;
an effector for manipulating said environment;
and
means for performing autonomous learning in
20 response to reinforcement from said environment.
7. An autonomous software agent as in claim 6, wherein said environment comprises a software environment or a physical environment.
8. An autonomous software agent as in claim 6,
25 wherein said environment comprises a carpenter's bench, a chess program, a customer service system, a graphic database, a help-desk system, a knowledge discovery system, a loan portfolio database, a music audio database,

an oil refinery, a piece of office equipment, or a physical room.

9. An autonomous software agent as in claim 6, wherein said agent is implemented in a case-based reasoning system.

10. An autonomous software agent for operating in an environment, comprising
a sensor for gathering information from said environment;
10 an effector for manipulating said environment;
a case base having a plurality of exemplar cases;
an inference engine capable of performing case-based reasoning steps on said cases; and
15 means for altering said case base in response to reinforcement from said environment.

11. An autonomous software agent as in claim 10, wherein said reinforcement comprises an evaluation of said agent.

20 12. An autonomous software agent for operating in an environment, comprising
a sensor for gathering information from said environment;
an effector for manipulating said environment;
25 a case base having a plurality of exemplar cases, said case base having a memory of cases, at least part of said memory being determined by a genetic technique;
an inference engine capable of performing case-based reasoning steps on said cases; and
30 means for altering said case base in response to reinforcement from said environment.

13. An autonomous software agent as in claim 12, wherein said reinforcement comprises an evaluation of said agent.

14. An autonomous software agent as in claim 12,
5 wherein said memory has a predetermined maximum size.

15. An autonomous software agent as in claim 12, wherein said memory comprises a case which has never been encountered in said environment.

16. An autonomous software agent as in claim 12,
10 wherein said genetic technique comprises the steps of
generating a case which has not been encountered
in said environment;

evaluating a case in response to (a) a set of
matching cases, or (b) said reinforcement; and

15 selecting a limited set of cases which provides
a preferred model of said environment.

17. An autonomous software agent as in claim 12, comprising means for selecting a case for case-based reasoning based on a plurality of measures.

20 18. An autonomous software agent as in claim 17, wherein said a plurality of measures comprises match quality, case accuracy, or case utility.

19. An autonomous software agent as in claim 12, comprising means for selecting a case for case-based
25 reasoning which may which may induce experimentation by
the autonomous agent.

20. An autonomous software agent as in claim 19, wherein said means for selecting a case comprises means for generating a random or pseudorandom effect.

21. An autonomous software agent as in claim 19, wherein said means for selecting a case comprises means for inducing experimentation by said agent.

22. An autonomous software agent as in claim 19,
5 wherein said means for selecting an action for the purpose of solving a problem and for selecting an action for the purpose of gathering further information so as to better solve problems.

23. An autonomous software agent as in claim 12,
10 comprising means for cooperating with a second autonomous software agent in selecting an action to be performed.

24. An autonomous software agent for operating in an environment, comprising
a receptor coupled to said environment and
15 generating a features message;
an inference engine coupled to said receptor, said inference engine comprising a case base, means for selecting a set of matching cases from said case base, means for altering said case base by means of a genetic
20 technique in response to said features message and in response to a reinforcement from said environment, and means for generating a commands message in response to said set of matching cases; and
an effector coupled to said environment and
25 operating on said environment in response to said commands message.

25. An autonomous software agent as in claim 24, wherein said means for altering comprises
means for generating a case which has not been
30 encountered in said environment;
means for evaluating a case in response to (a) a set of matching cases, or (b) said reinforcement; and

means for selecting a limited set of cases which provides a preferred model of said environment.

26. An autonomous software agent as in claim 24, wherein said means for selecting a set of matching cases
5 from said case base comprises a random effect or a pseudorandom effect.

27. A case-based reasoning system, comprising
a case base having a plurality of exemplar cases; and
10 an inference engine capable of performing case-based reasoning steps on said cases;
said case base having been constructed substantially by a genetic technique.

28. A case-based reasoning system as in claim 27,
15 wherein said system is capable of operating in an environment, and wherein said genetic technique comprises the steps of

generating a case which has not been encountered in said environment;
20 evaluating a case in response to (a) a set of matching cases, or (b) a reinforcement received from said environment; and

selecting a limited set of cases which provides a preferred model of said environment.

29. A case-based reasoning system as in claim 28,
25 wherein said step of generating comprises the steps of inspecting a scenario encountered in said environment;

altering a case in said case base to form a new
30 case; or

combining a first aspect of a first case in said case base with a second aspect of a second case in said case base to form a new case.

30. A case-based reasoning system, comprising
a case base having a plurality of exemplar
cases; and

an inference engine capable of performing case-
5 based reasoning steps on said cases;

said case base including a case which has not
been encountered in an environment and which has not been
entered from an external interface.

31. A case-based reasoning system, comprising
10 a case base having a plurality of exemplar
cases; and

an inference engine capable of performing case-
based reasoning steps on said cases;

said case base comprising a predetermined
15 maximum number of cases and having been constructed
substantially by a genetic technique.

32. A case-based reasoning system as in claim 31,
comprising means for selecting a set of matching cases
from said case base having a random effect or a
20 pseudorandom effect.

33. A case-based reasoning system as in claim 32,
wherein said means for selecting a set of matching cases
from said case base which applies a random effect or a
pseudorandom effect to a measure of match quality, case
25 accuracy, or case utility.

34. A case-based reasoning system as in claim 32,
wherein said means for selecting a set of matching cases
from said case base is more likely to select a case with
a greater measure of match quality, case accuracy, or case
30 utility.

35. A case-based reasoning system as in claim 32,
wherein said means for selecting a set of matching cases

from said case base has a likelihood for selecting a first case over a second case in linear proportion to a ratio of a measure of case accuracy of said first case over said second case.

5 36. A method of operating a software agent, comprising the step of performing autonomous learning.

37. A method as in claim 36, comprising the step of collecting a case base of experience in an environment.

38. A method as in claim 36, comprising the step of
10 altering said case base in response to an evaluation of said experience.

39. A method as in claim 36, comprising the step of responding to reinforcement from an environment.

40. A method of operating an autonomous software
15 agent in an environment, comprising the steps of

generating a features message in response to said environment;

altering a case base of experience in said environment by means of a genetic technique in response to
20 said features message and in response to a reinforcement from said environment;

selecting a set of matching cases from said case base;

generating a commands message in response to
25 said set of matching cases; and

operating on said environment in response to said commands message.

41. An autonomous software agent as in claim 40, wherein said step of altering comprises

30 generating a case which has not been encountered in said environment;

evaluating a case in response to (a) a set of matching cases, or (b) said reinforcement; and
selecting a limited set of cases which provides a preferred model of said environment.

5 42. A method of operating a case-based reasoning system, comprising the steps of
 constructing a case base having a plurality of exemplar cases substantially by a genetic technique; and
 performing case-based reasoning steps on said
10 cases.

 43. A method as in claim 42, wherein said genetic technique comprises the steps of
 generating a case which has not been encountered in an environment;
15 evaluating a case in response to (a) a set of matching cases, or (b) a reinforcement received from said environment; and
 selecting a limited set of cases which provides a preferred model of said environment.

20 44. A method as in claim 43, wherein said step of generating comprises the steps of
 inspecting a scenario encountered in said environment;
 altering a case in said case base to form a new
25 case; or
 combining a first aspect of a first case in said case base with a second aspect of a second case in said case base to form a new case.

 45. In a case-based reasoning system, a method of
30 altering a case base having a plurality of exemplar cases; said method comprising
 generating a case which has not been encountered in said environment;

evaluating a case in response to (a) a set of matching cases, or (b) a reinforcement received from said environment; and

5 selecting a limited set of cases which provides a preferred model of said environment.

46. In a case-based reasoning system, a case base which has been selected by a genetic technique.

47. A case base as in claim 46, said genetic technique comprising

10 generating a case which has not been encountered in said environment;

evaluating a case in response to (a) a set of matching cases, or (b) a reinforcement received from said environment; and

15 selecting a limited set of cases which provides a preferred model of said environment.

48. In a case-based reasoning system, a case base having a plurality of exemplar cases, substantially all said cases comprising

20 a set of matchable features;
an action to be taken when said case is selected; and
a measure of value for said case.

49. A case base as in claim 48, wherein said measure
25 of value comprises match quality, case accuracy, or case utility.

50. A case base as in claim 48, wherein substantially all said cases comprise a second measure of value for said case.

30 51. A system for operating in an environment, comprising

a receptor coupled to said environment and generating a features message;

a first inference engine coupled to said features message and a reinforcement from said environment, for generating a commands message;

a second inference engine coupled to said features message and said reinforcement from said environment, for altering said commands message; and

an effector coupled to said environment and operating on said environment in response to said commands message.

52. A system for operating in an environment, comprising

a receptor coupled to said environment and generating a features message;

a first inference engine comprising a first case base, first means for selecting a first set of matching cases from said first case base, means for altering said first case base by means of a genetic technique in response to said features message and in response to a reinforcement from said environment, and means for generating a commands message in response to said first set of matching cases;

a second inference engine comprising a second case base, second means for selecting a set of matching cases from said second case base, means for altering said second case base by means of a genetic technique in response to said commands message, in response to said features message and in response to said reinforcement from said environment, and means for altering said commands message in response to said second set of matching cases; and

an effector coupled to said environment and operating on said environment in response to said commands message.

53. A method for operating in an environment, comprising

generating a features message in response to said environment, said features message having a set of
5 features;

first matching said set of features to a first case base;

generating a commands message in response to said first step of matching and a reinforcement from said
10 environment;

second matching said set of features to a second case base;

altering said commands message in response to said second step matching and said reinforcement from said
15 environment; and

operating on said environment in response to said commands message.

54. A method for operating in an environment, comprising

20 generating a features message in response to said environment;

selecting a first set of matching cases from a first case base;

altering said first case base by means of a
25 genetic technique in response to said features message and in response to a reinforcement from said environment;

generating a commands message in response to said first set of matching cases;

30 selecting a second set of matching cases from a second case base;

altering said second case base by means of a genetic technique in response to said commands message, in response to said features message and in response to said reinforcement from said environment;

35 altering said commands message in response to said second set of matching cases; and

operating on said environment in response to
said commands message.

1/7

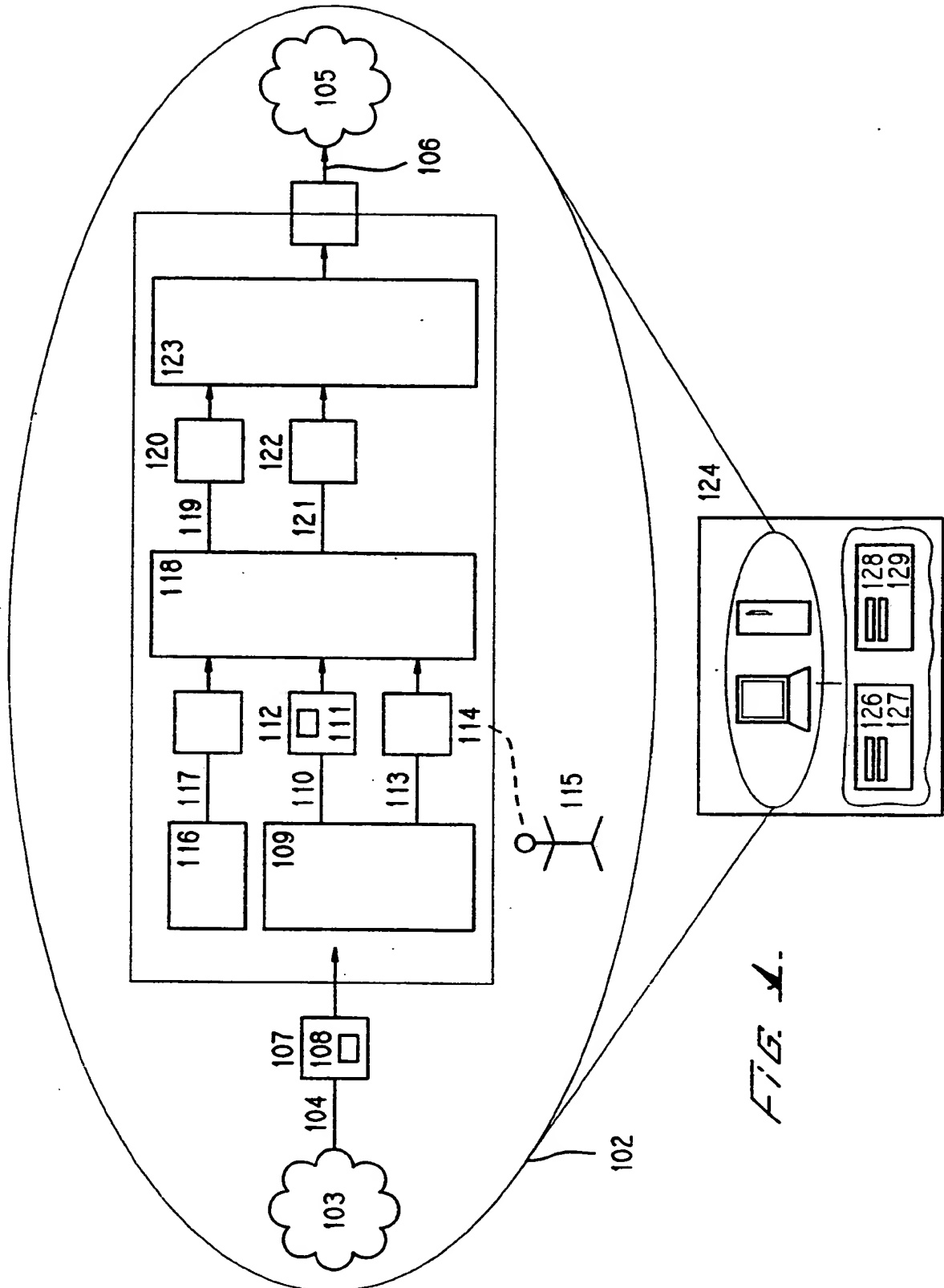


FIG. 1

SUBSTITUTE SHEET

2 / 7

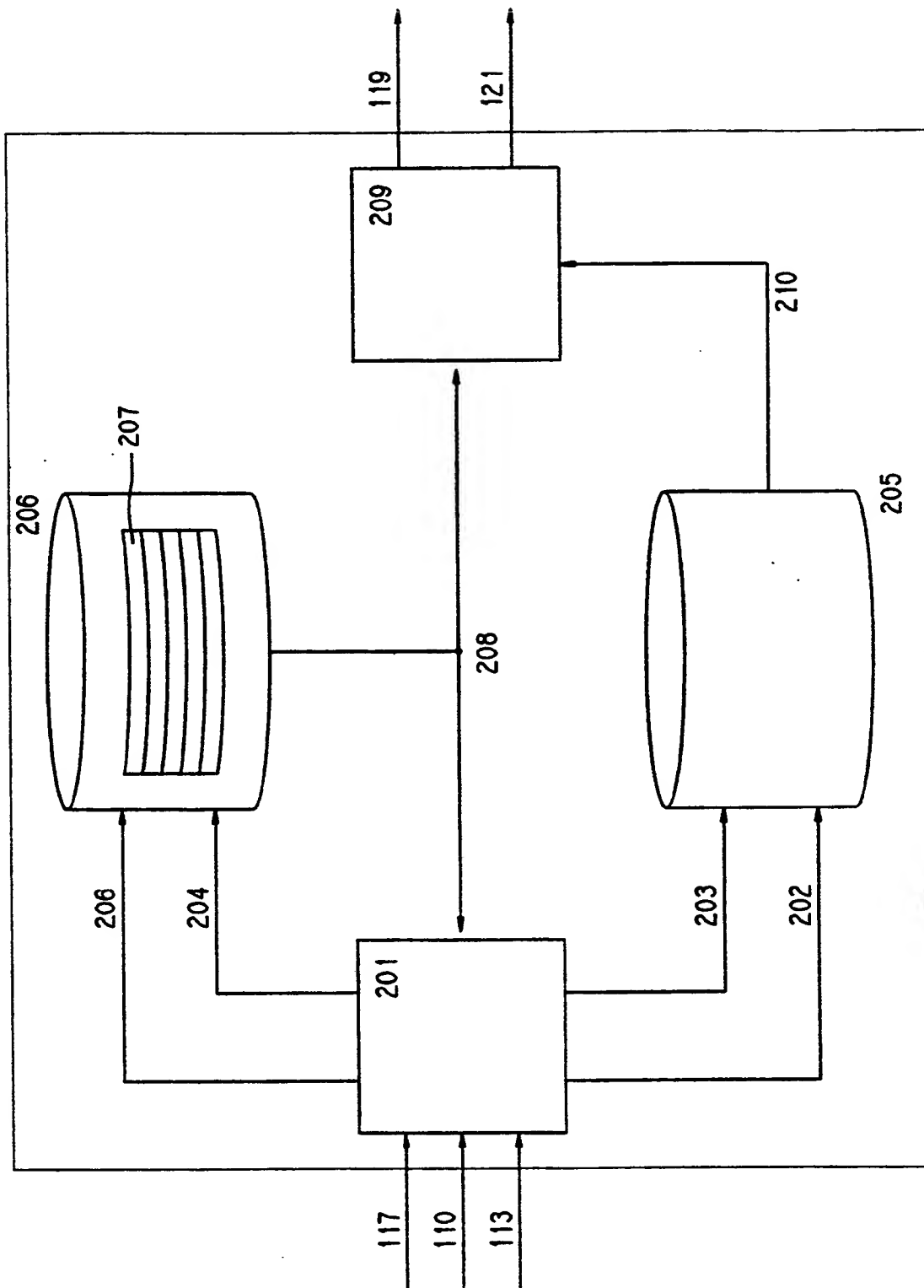
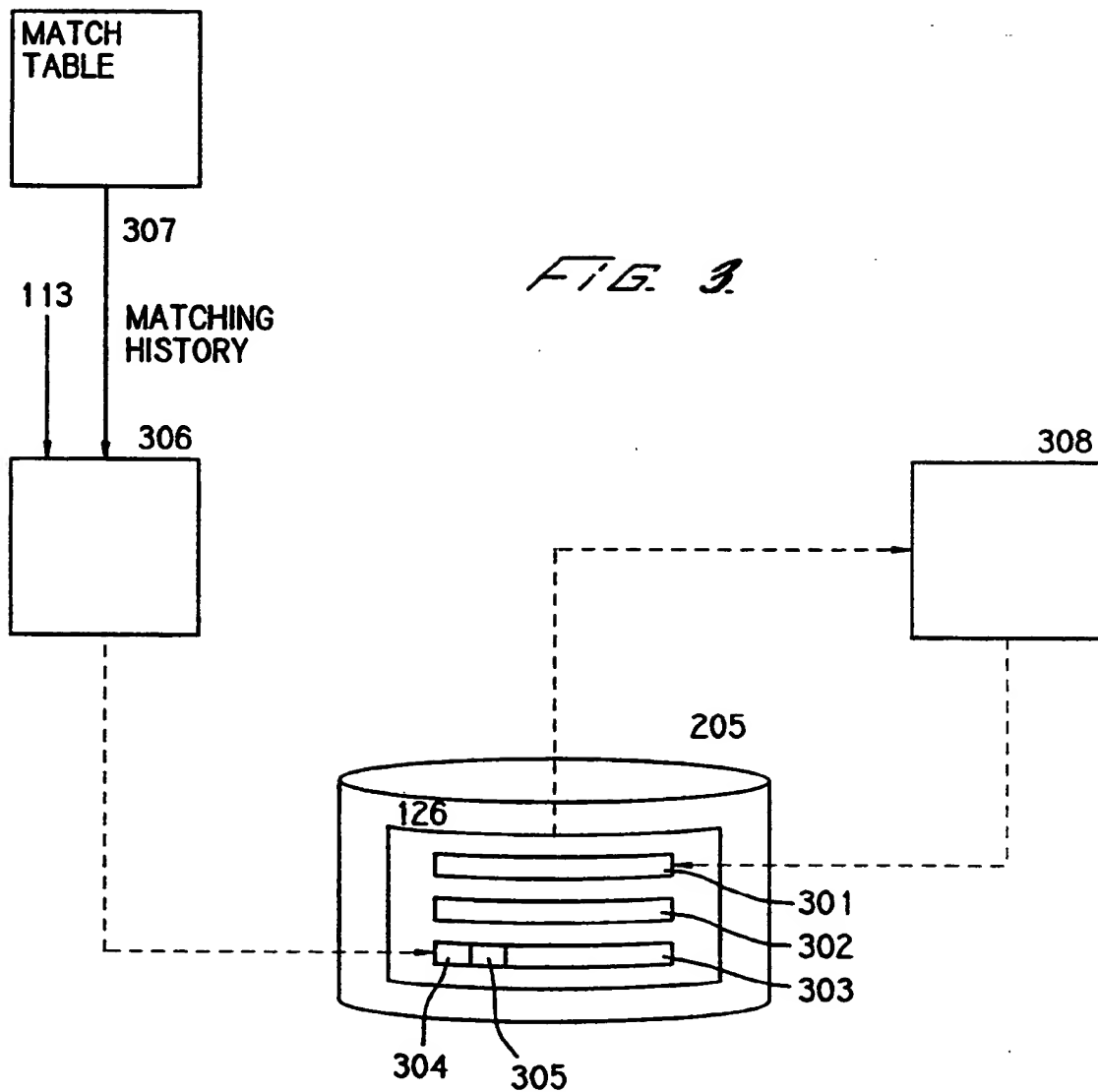


FIG. 2.

SUBSTITUTE SHEET

3/7



SUBSTITUTE SHEET

4 / 7

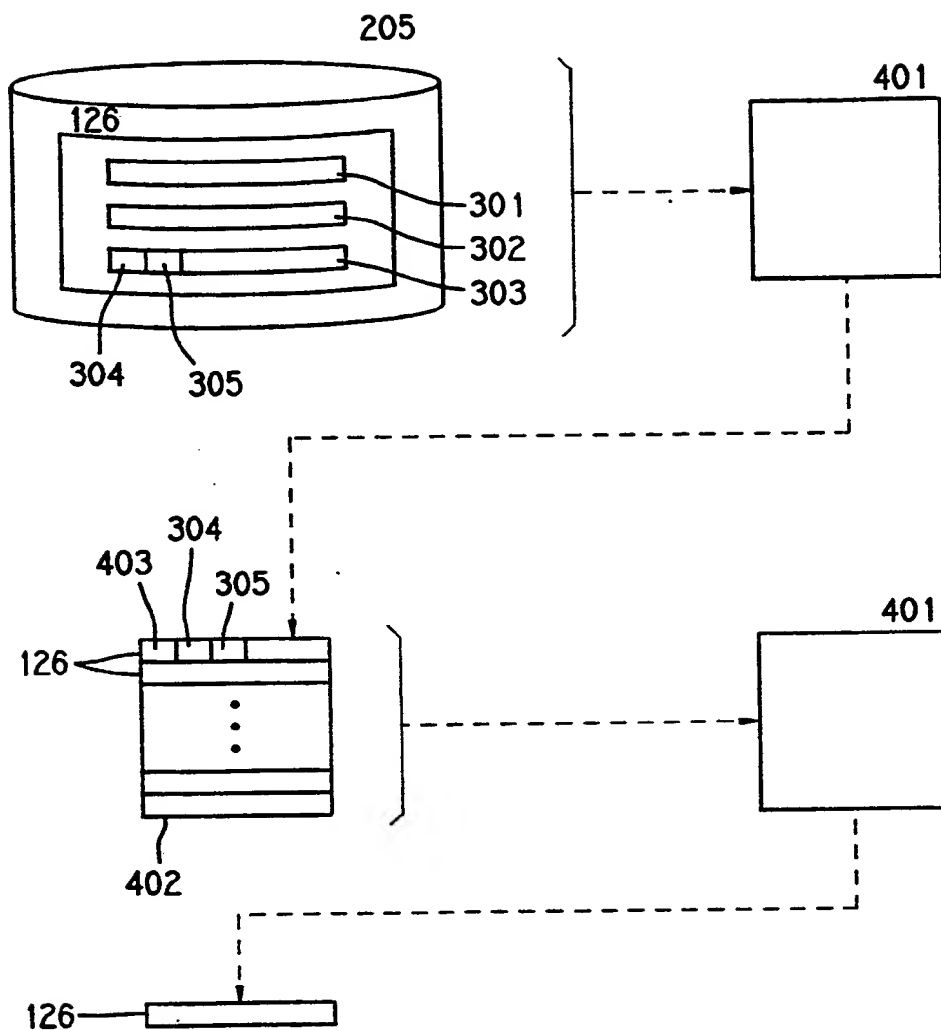
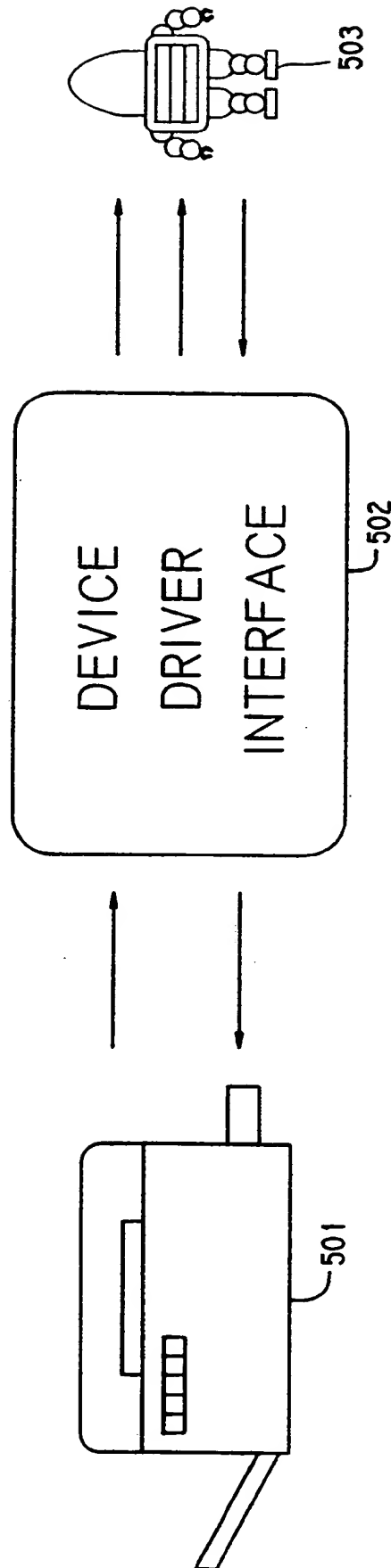


FIG. 4

SUBSTITUTE SHEET

5 / 7

KNOBOTS IN DIAGNOSIS AND REPAIR



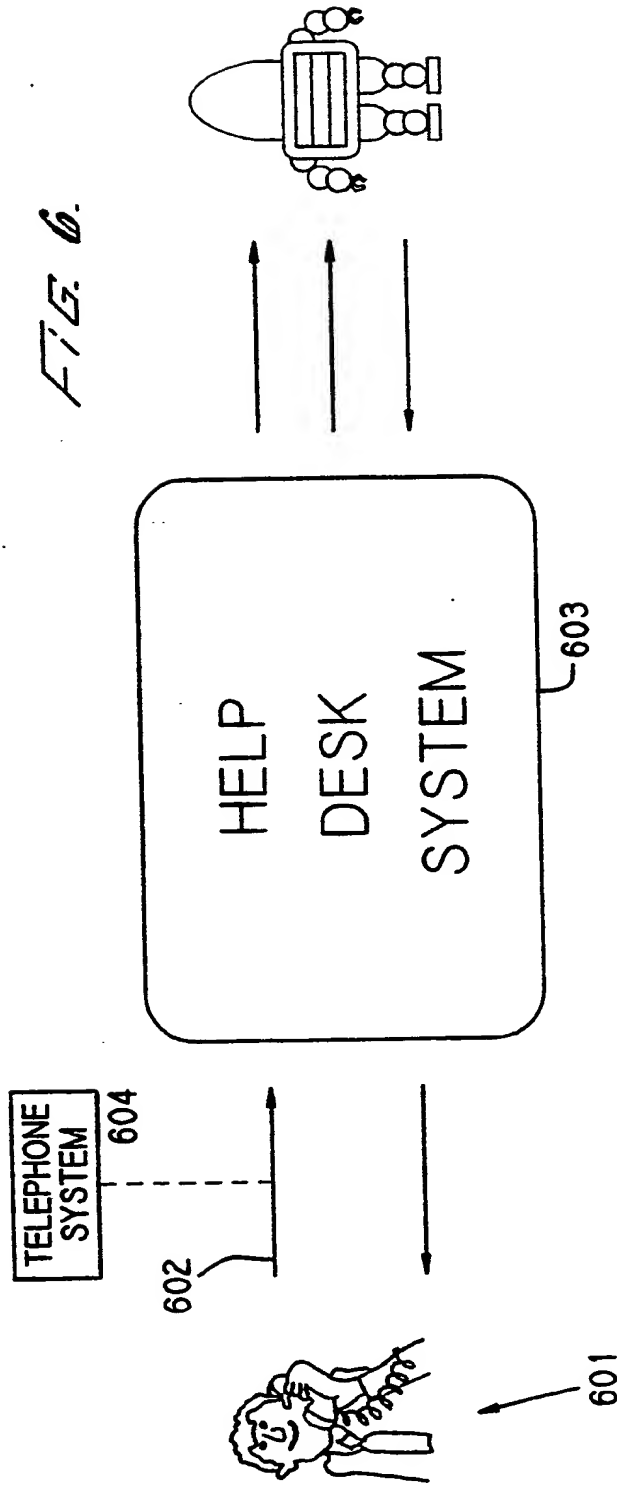
GOALS	ELIMINATE NEED FOR USER INVOLVEMENT IN MAINTENANCE. ANTICIPATE FAILURES FOR PREVENTIVE MAINTENANCE
STIMULI	DEVICE SENSOR READINGS, USER AND DEALER COMMANDS
ACTIONS	DEVICE DRIVER OUTPUTS, DIAGNOSTIC AND REPAIR MESSAGES
REINFORCEMENTS	NEGATIVE IF DEVICE FAILS
BENEFITS	GREATLY INCREASED MEAN-TIME-BETWEEN-FAILURE

FIG. 5.

SUBSTITUTE SHEET

6 / 7

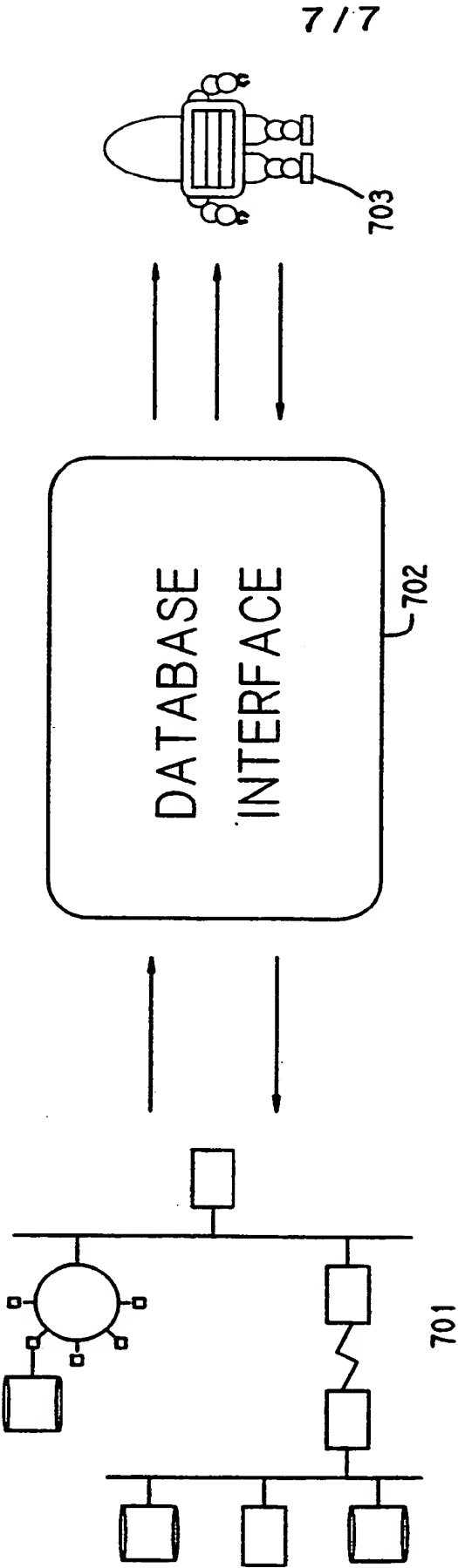
Knobots in Customer Service



GOALS	RESPONSIVELY HANDLE CUSTOMER PHONE CALLS AND AUTOMATICALLY RESOLVE PROBLEM
STIMULI	TOUCH-TONE COMMANDS, VOICE RESPONSE UNIT MESSAGES
ACTIONS	VOICE RESPONSE OUTPUT, PBX TRANSFERS
REINFORCEMENTS	POSITIVE IF DIALOG COMPLETED, NEGATIVE IF CALLER HANGS UP BEFORE PROBLEM RESOLUTION OR TRANSFER
BENEFITS	FOCUSED, RESPONSIVE USER INTERFACE THAT ADAPTS TO USER BEHAVIOR

SUBSTITUTE SHEET

KNOBOTS IN KNOWLEDGE DISCOVERY



GOALS	LEARN USEFUL KNOWLEDGE ABOUT PATTERNS IN DATA
STIMULI	DATABASE TABLES AND SCHEMA DEFINITIONS
ACTIONS	DATABASE QUERIES AND TRANSACTIONS
REINFORCEMENTS	POSITIVE IF USEFUL INFORMATION DISCOVERED
BENEFITS	IMPROVED ABILITY TO GENERALIZE FROM DATA AND CAPTURE USEFUL REGULARITIES FOR COMPETITIVE ADVANTAGE

FIG. 1

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US93/03557**A. CLASSIFICATION OF SUBJECT MATTER**

IPC(5) :G06F 15/18

US CL :395/10, 13

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 395/10, 13

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS search terms: case based

DIALOG searched terms: case base?; genetic algorithm?

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X,P --- Y	US, A, 5,136,686 (KOZA) 04 AUGUST 1992, ABSTRACT.	1-9 AND 36-39 ----- 10-35, 40, 41
X,E --- Y	US, A, 5,224,206 (SIMOUDIS) 29 JUNE 1993 ABSTRACT AND COLUMN 4, LINES 4-13.	1-9 AND 36-39 ----- 10-35, 40, 41
A	Proceedings of the Workshop on Case-Based Reasoning; 10-13 May 1988; Phyllis Koton; "REASONING ABOUT EVIDENCE IN CAUSAL EXPLANATIONS"; Pages 260-270.	1-54

☐ Further documents are listed in the continuation of Box C.
 ☐ See patent family annex.

* Special categories of cited documents:	T	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
A document defining the general state of the art which is not considered to be part of particular relevance	X*	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
E earlier document published on or after the international filing date	Y*	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
L document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	&*	document member of the same patent family
O document referring to an oral disclosure, use, exhibition or other means		
P document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search

06 August 1993

Date of mailing of the international search report

20 SEP 1993

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231

Facsimile No. NOT APPLICABLE

Authorized officer

ALLEN MACDONALD
Allen Macdonald
Telephone No. (703) 305-9600

Form PCT/ISA/210 (second sheet)(July 1992)*

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US93/03557

Box I Observations where certain claims were found unsearchable (Continuation of item 1 of first sheet)

This international report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box II Observations where unity of invention is lacking (Continuation of item 2 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

Please See Extra Sheet.

1. ☒ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
2. ☐ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.
3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:
...
4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐

The additional search fees were accompanied by the applicant's protest.

☒

No protest accompanied the payment of additional search fees.

Form PCT/ISA/210 (continuation of first sheet(1))(July 1992)*

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US93/03557

BOX II. OBSERVATIONS WHERE UNITY OF INVENTION WAS LACKING

This ISA found multiple inventions as follows:

Group I, claims 1-9 and 36-39, drawn to a learning Autonomous Software agent, classified in class 364, subclass 424.02.

Group II, claims 10-26 and 40-41, drawn to an alterable case based Autonomous Software Agent, classified in class 395, subclass 77.

Group III, claims 27-35, drawn to a case based reasoning system, classified in class 395, subclass 50.

Group IV, claims 42-44, drawn to genetically constructing a case based, classified in class 395, subclass 13.

Group V, claim 45, drawn to altering a case base, classified in class 395, subclass 51.

Group VI, claims 46-47, drawn to a first case base, classified in class 395, subclass 60.

Group VII, claims 48-50, drawn to a second case base, classified in class 395, subclass 60.

Group VIII, claim 51, drawn to a first plural processing device system, classified in class 395, subclass 11.

Group IX, claims 52 and 54 drawn to a second plural processing device system, classified in class 395, subclass 11.

Group X, claim 53, drawn to a third plural processing device system, classified in class 395, subclass 11.